VTune™ Performance Analyzer 4.0

A comprehensive solution for developers of high performance software for the Intel® Pentium® III processor is available.

Product Highlights

- A Quick Performance Analysis wizard to help you quickly find the performance bottlenecks in your code.
- Call graph profiling to profile your Windows* or Java* application, generate a call graph, and identify critical functions and call-sequences.
- Non-intrusive monitoring to provide system-wide view of software activity, including that of your application and operating system.
- Hotspot analysis to help you identify potential performance problems in modules, classes, and functions.
- Tuning coach for source level tuning advice for multiple processors, including the Pentium III processor, and multiple languages such as C/C++, Java, FORTRAN, and assembly language.
- Static assembly analysis to let you examine how your program executes on the latest Intel architecture, including the Pentium III processor.

What's new in 4.0?

Performance analysis for the latest Intel® Pentium® III processor

You can use sampling, call graph profiling, or assembly analysis to analyze the performance of your application on the Pentium III processor and find potential performance bottlenecks. You can select a Pentium III processor-specific event and run a sampling session based on this event.

Tuning advice for the Streaming SIMD Extensions

You can use the coach to analyze performance bottlenecks and get tuning advice specific to the Pentium III processor. The coach analyzes your code and, whenever appropriate, to boost the performance of your program, suggests intrinsics corresponding to the Streaming SIMD Extensions of the Pentium III processor.

Native compilation for Java* methods

You can use the bytecode accelerator to analyze selected methods in your Java class file, identify critical, time-intensive methods that would benefit from native compilation, and compile and optimize those methods to native Intel architecture code.

Improved chronology-collection interface

You can use the improved chronology-collection interface to collect processor and operating system counter data, independent of sampling. Track an operating system or processor object, and the performance of its subsystem to see how your application affects the performance of other subsystems within your computer and viceversa.

Event ratios

You can use event ratios to obtain detailed performance information on combinations of processor events, such as Instructions Retired per Cycle and Branch Mispredictions per Branch Retired, that were monitored and sampled in multiple event-based sampling sessions.

Call-site information

Using call graph profiling, you can collect and display call-site information in the call list and source views. Call site information includes the number of times a function is called from a specific location and the time it took.

Enhanced source view

In the source code view, you can combine and display call graph data and sampling data from multiple-event sessions. For example, for a specific function in the source view, you can display the number of samples collected for the function for each event it was monitored on, the number of times the function was called from different locations, and the time spent in each location.

Online Help for the Pentium III Processor

You can isplay context-sensitive online descriptions of the Pentium III Processor's new instructions and corresponding intrinsics.

How can you analyze the performance of your program with the VTune analyzer?

- Use Quick Performance Analysis or Project Wizard to specify the Windows or Java application to analyze.
- Run a non-intrusive sampling session. Monitor your application along with the rest of the software on your system, analyze its performance, and display the performance bottlenecks in your module, functions, and source code.
- Run a call graph profiling session. Profile all the functions or JIT-compiled Java methods loaded into memory, analyze the data, and generate a call graph showing all called functions, their parent and child functions, the number of times each function was called, and time spent in each call.
- From the call graph profiling or sampling analysis views, identify the critical function, and, using a simple point-and-click interface, zoom right into the source code view of the function. Identify performance issues associated with each instruction within the function and invoke the code coach or assembly coach for source-level tuning advice.
- Track operating system objects and counters, such as page faults per second, and inspect chronology graphs for each counter.
- Get advice from the coach on how to use the Streaming SIMD extensions and MMX™ technology intrinsics to improve the performance of your program. You can use the Intel® C\C++ Compiler 4.0 to compile and optimize your code with these intrinsics.

Printed in USA/2K/0598/JM, Order Number: 244463-001